OIPE
APR 17 2007
PATENT & TRADEMARK OFFICE

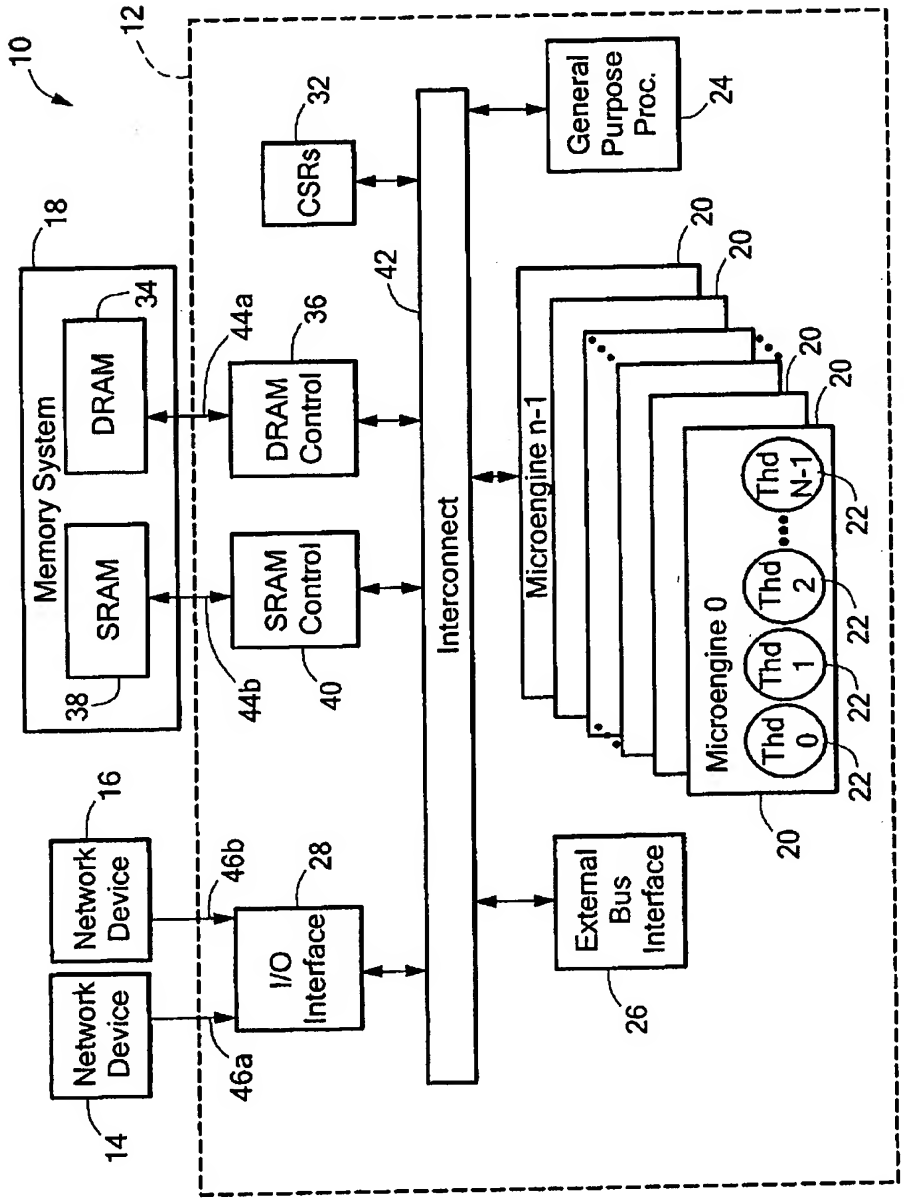

FIG. 1

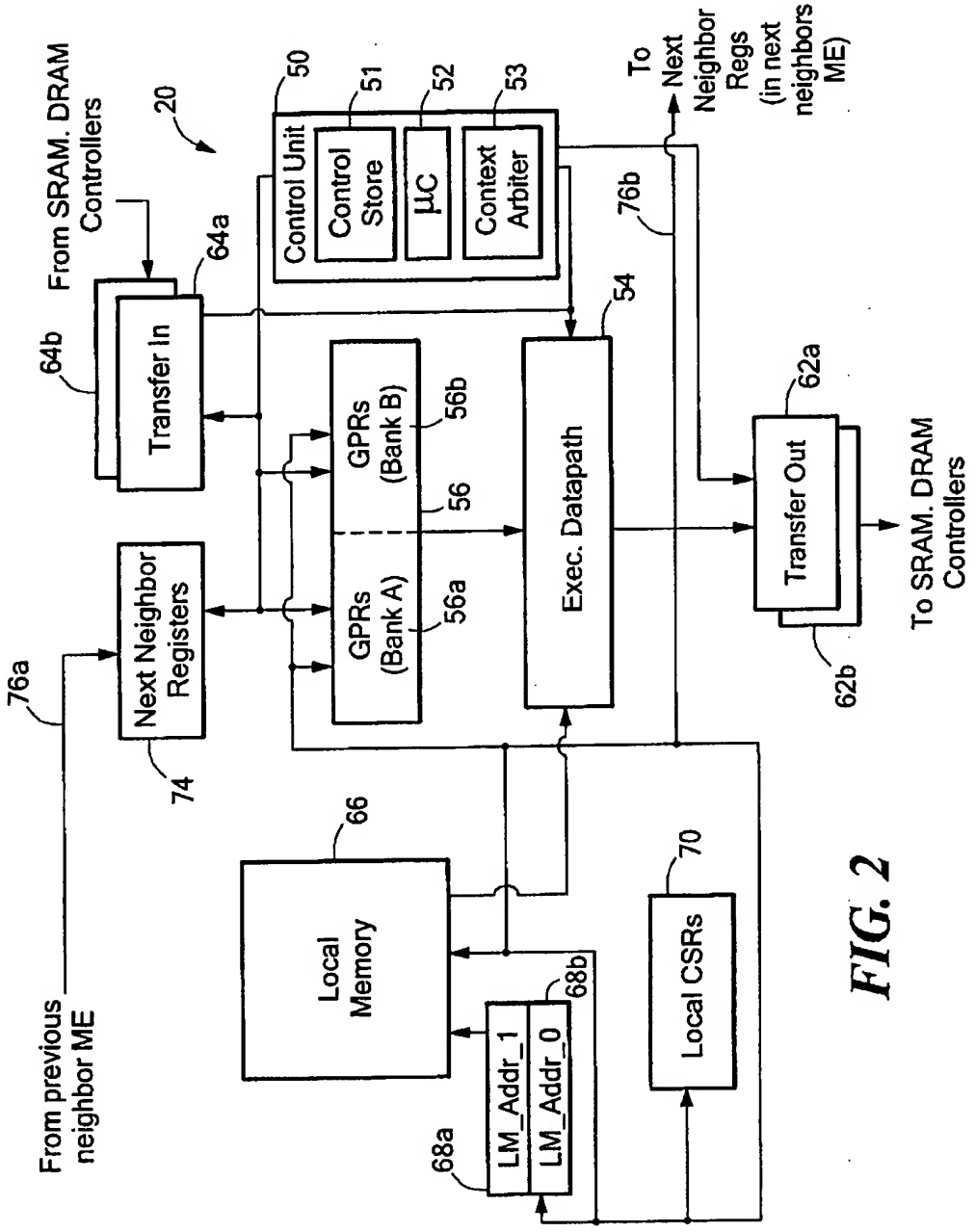
From SRAM. DRAM Controllers
20
50
Control Unit
51
Control Store
52
µC
53
Context Arbiter
To Next Neighbor Regs (in next neighbors ME)
64a
64b
Transfer In
54
76b
GPRs (Bank B)
56b
56
GPRs (Bank A)
56a
Exec. Datapath
62a
Transfer Out
To SRAM. DRAM Controllers
62b
76a
Next Neighbor Registers
74
66
Local Memory
70
Local CSRs
68b
LM_Addr_1
LM_Addr_0
68a
From previous neighbor ME

FIG. 2

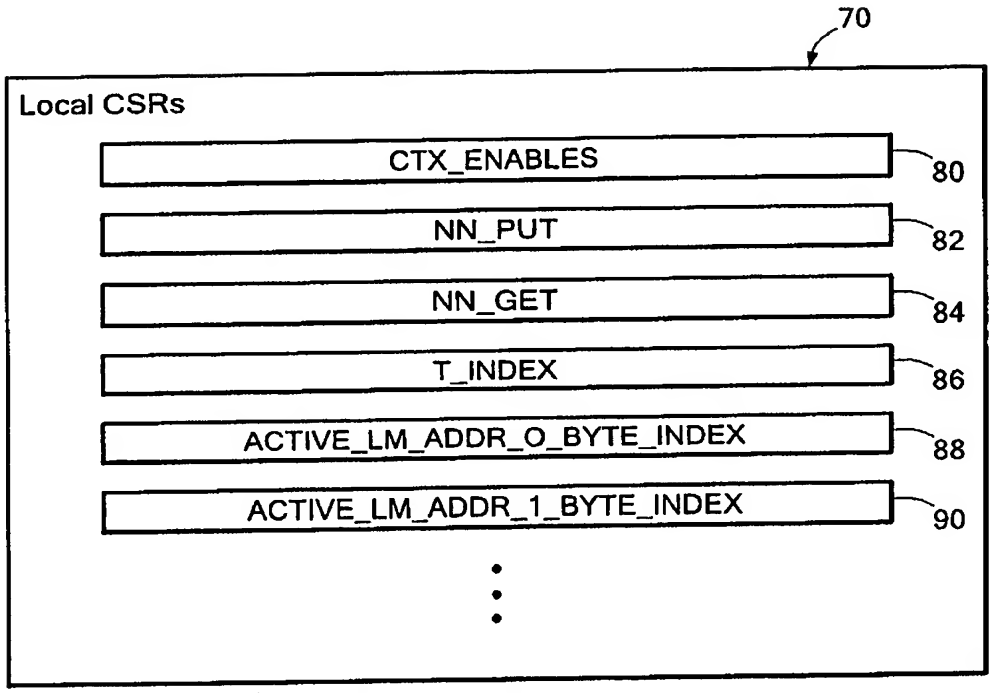
70
Local CSRs
CTX_ENABLES
80
NN_PUT
82
NN_GET
84
T_INDEX
86
ACTIVE_LM_ADDR_O_BYTE_INDEX
88
ACTIVE_LM_ADDR_1_BYTE_INDEX
90

FIG. 3

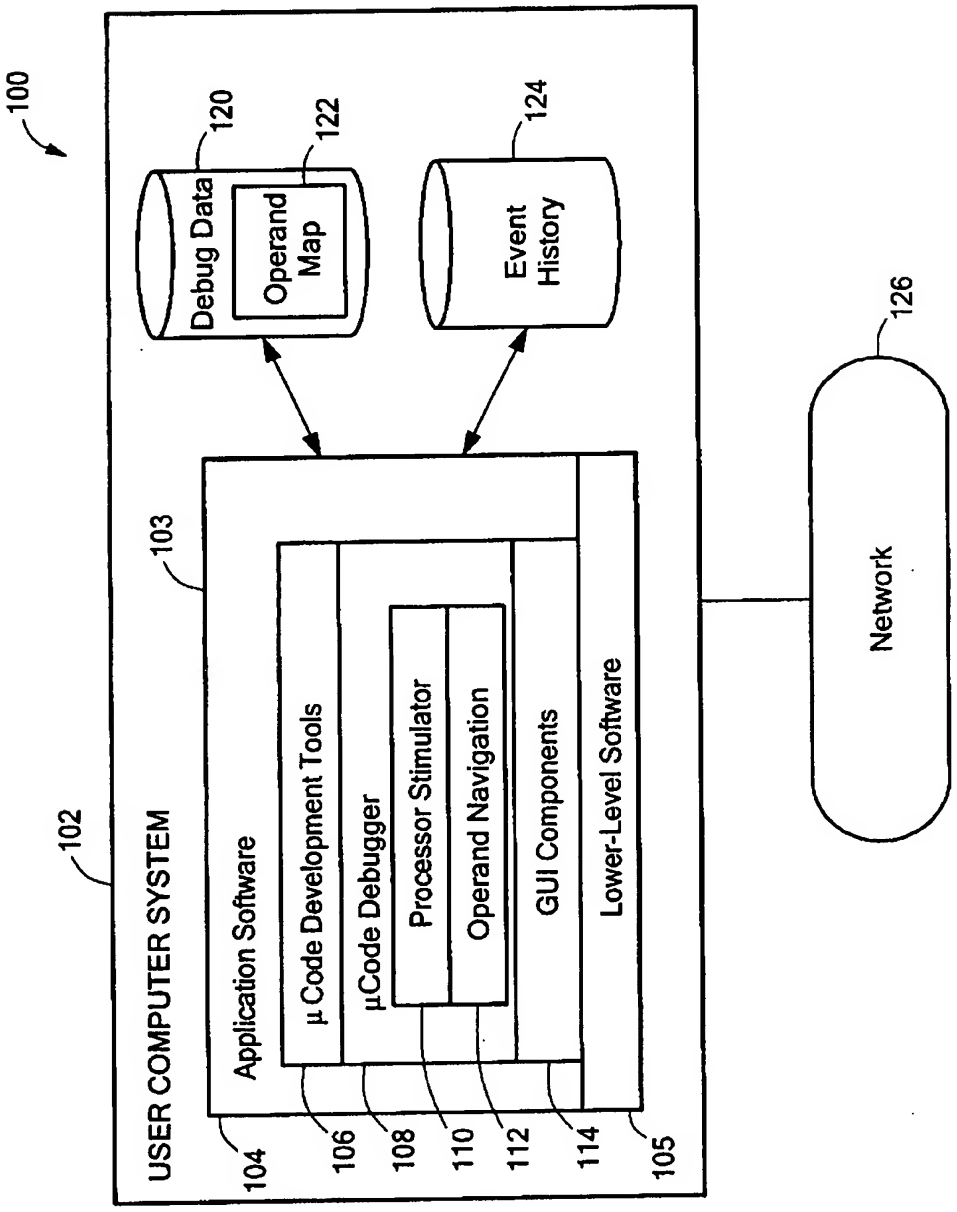
100
102
USER COMPUTER SYSTEM
103
Application Software
µ Code Development Tools
µCode Debugger
Processor Stimulator
Operand Navigation
GUI Components
Lower-Level Software
104
106
108
110
112
114
105
Debug Data
120
Operand Map
122
Event History
124
Network
126

FIG. 4

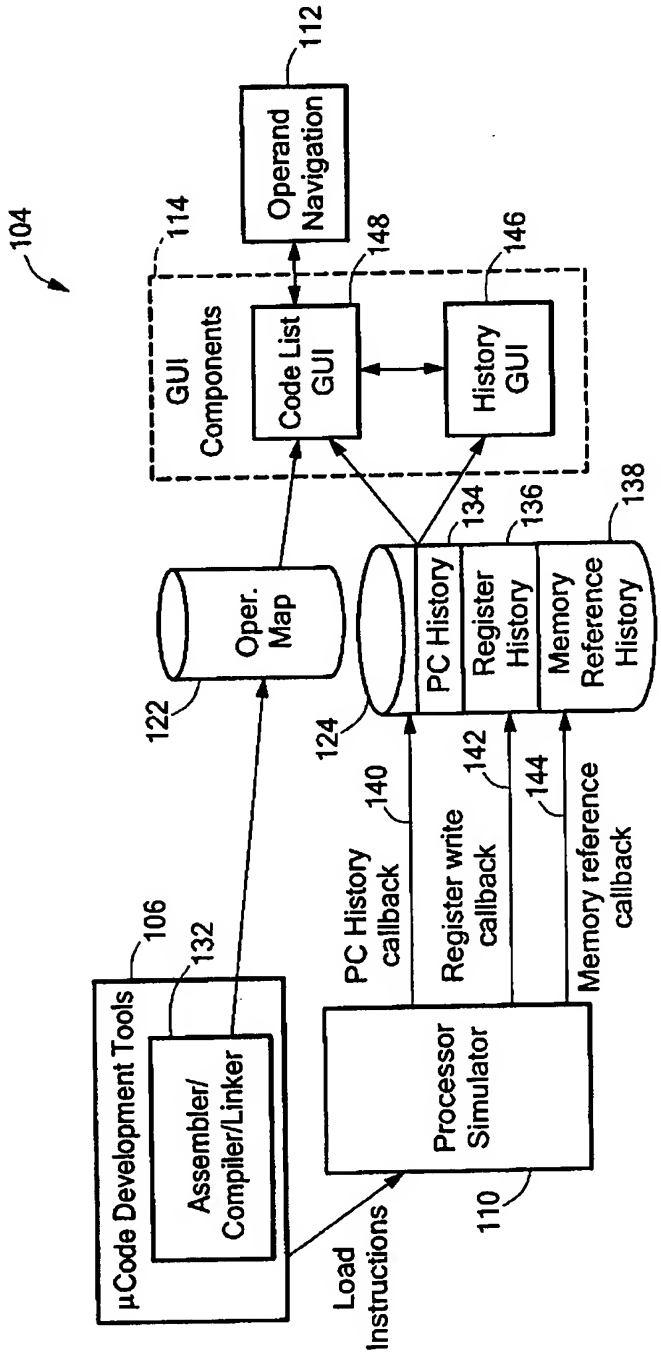
104
112
Operand Navigation
114
GUI Components
148
Code List GUI
146
History GUI
134
136
138
PC History
Register History
Memory Reference History
122
Oper. Map
124
140
142
144
PC History callback
Register write callback
Memory reference callback
106
132
µCode Development Tools
Assembler/ Compiler/Linker
Processor Simulator
110
Load Instructions

FIG. 5

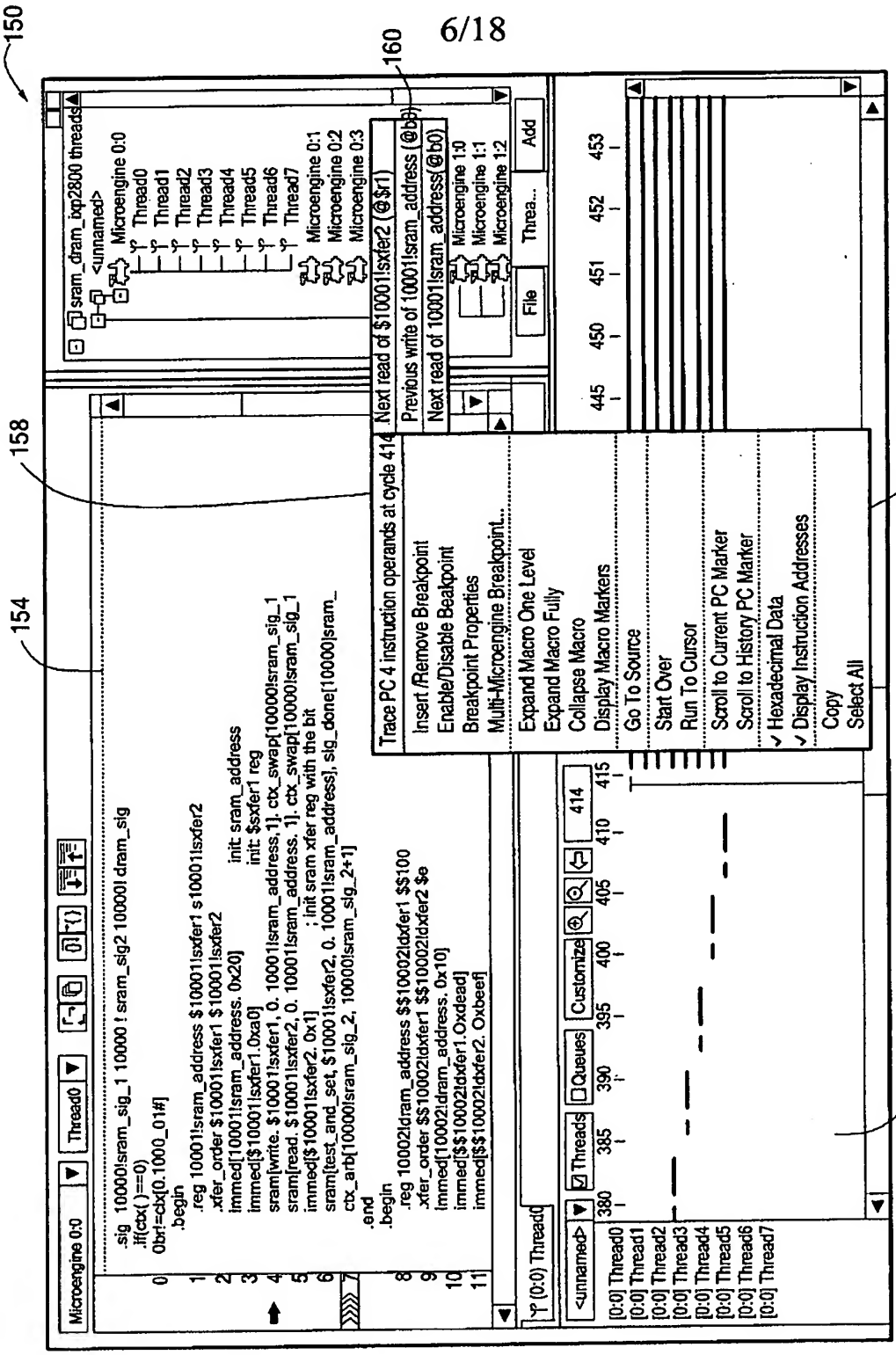
150
158
154
160
156
152
Microengine 0:0
Thread0
Queues
Customize
Threads
<unnamed>
Trace PC 4 instruction operands at cycle 414
Insert /Remove Breakpoint
Enable/Disable Beakpoint
Breakpoint Properties
Multi-Microengine Breakpoint...
Expand Macro One Level
Expand Macro Fully
Collapse Macro
Display Macro Markers
Go To Source
Start Over
Run To Cursor
Scroll to Current PC Marker
Scroll to History PC Marker
Hexadecimal Data
Display Instruction Addresses
Copy
Select All
Next read of $10001!sxfer2 (@$r1)
Previous write of 10001!sram_address (@b0)
Next read of 10001!sram_address(@b0)
sram_dram_ixp2800 threads
Microengine 1:0
Microengine 1:1
Microengine 1:2
File
Add

FIG. 6

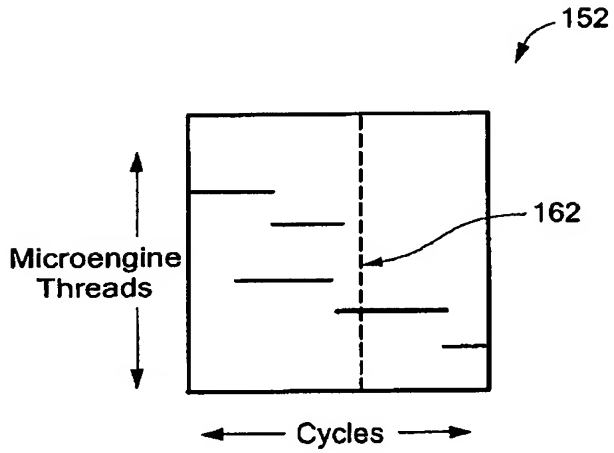
152
162
Microengine Threads
Cycles

FIG. 7

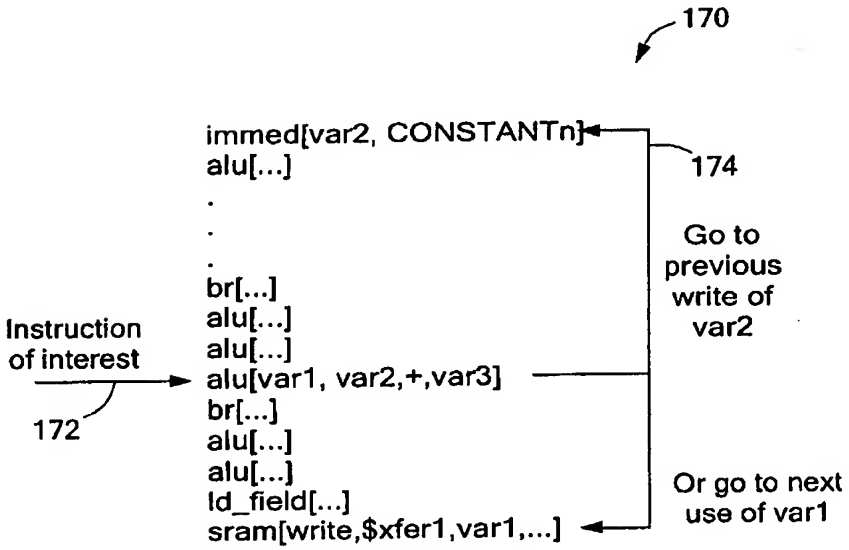
170
immed[var2, CONSTANTn]
alu[...]
.
.
.
br[...]
alu[...]
alu[...]
alu[var1, var2,+,var3]
br[...]
alu[...]
alu[...]
ld_field[...]
sram[write,$xfer1,var1,...]
174
Go to previous write of var2
Or go to next use of var1
Instruction of interest
172

FIG. 8

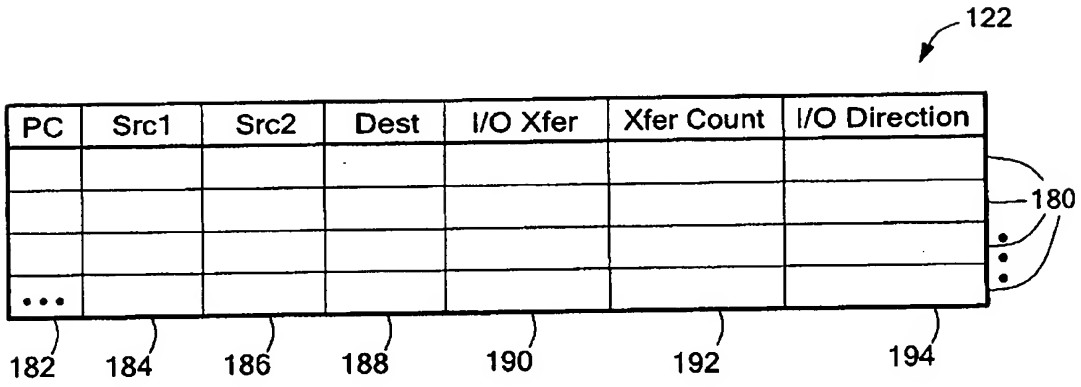
122
PC
Src1
Src2
Dest
I/O Xfer
Xfer Count
I/O Direction
180
182
184
186
188
190
192
194

FIG. 9

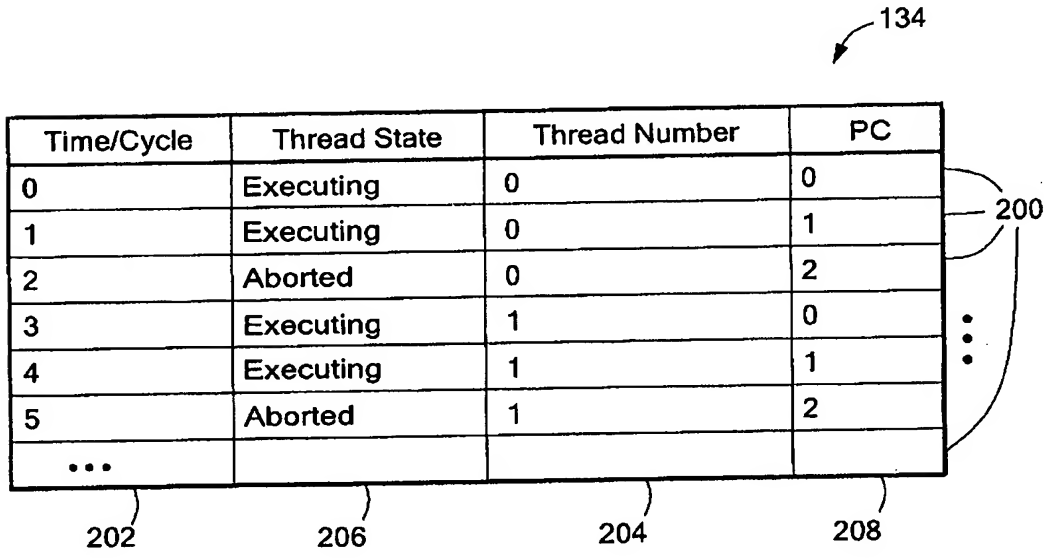
134
Time/Cycle
Thread State
Thread Number
PC
0 Executing 0 0
1 Executing 0 1
2 Aborted 0 2
3 Executing 1 0
4 Executing 1 1
5 Aborted 1 2
200
202
206
204
208

FIG. 10

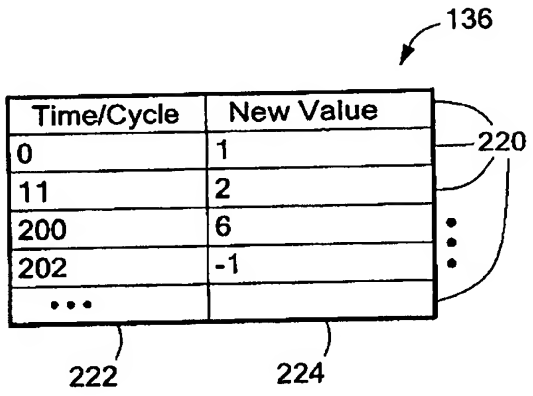
136
Time/Cycle
New Value
0
1
11
2
200
6
202
-1
220
222
224

FIG. 11

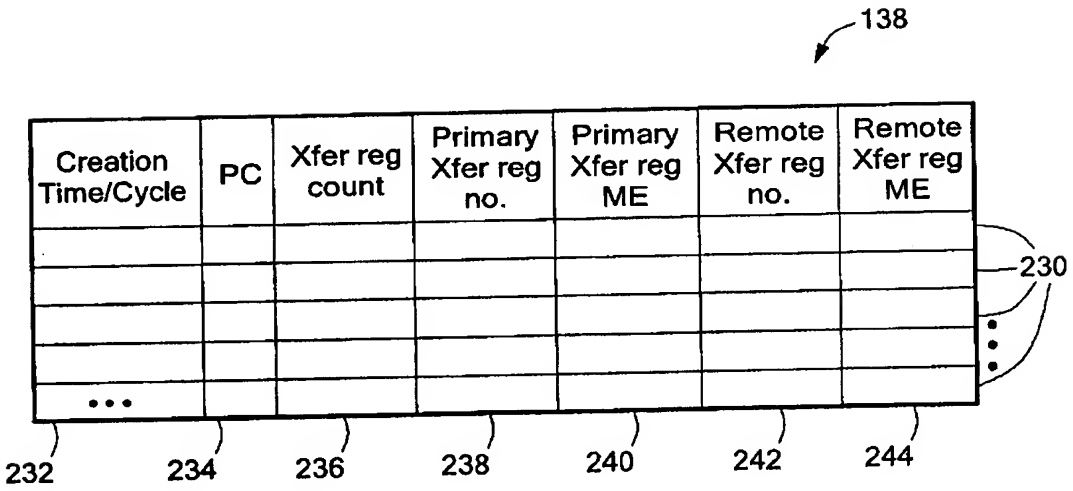
138
Creation Time/Cycle
PC
Xfer reg count
Primary Xfer reg no.
Primary Xfer reg ME
Remote Xfer reg no.
Remote Xfer reg ME
230
232
234
236
238
240
242
244

FIG. 12

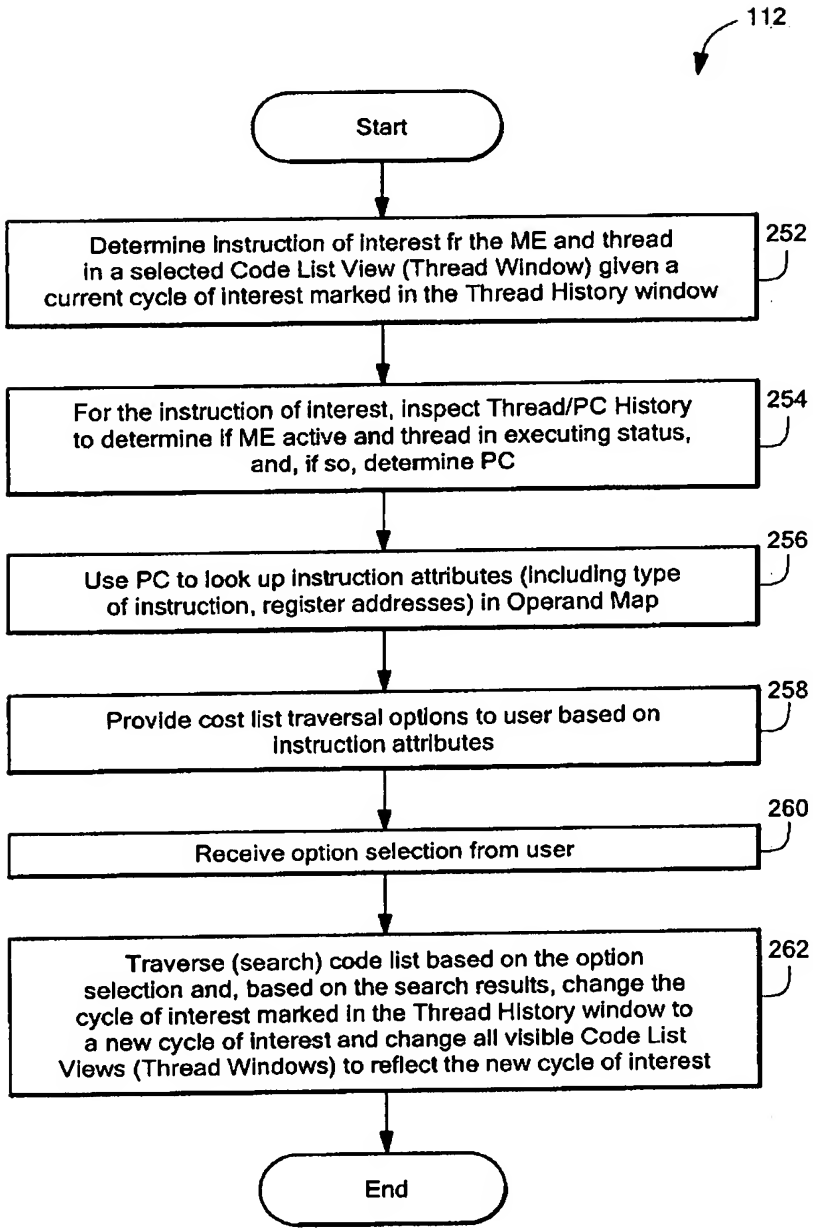
112
Start
Determine instruction of interest fr the ME and thread in a selected Code List View (Thread Window) given a current cycle of interest marked in the Thread History window
252
For the instruction of interest, inspect Thread/PC History to determine if ME active and thread in executing status, and, if so, determine PC
254
Use PC to look up instruction attributes (including type of instruction, register addresses) in Operand Map
256
Provide cost list traversal options to user based on instruction attributes
258
Receive option selection from user
260
Traverse (search) code list based on the option selection and, based on the search results, change the cycle of interest marked in the Thread History window to a new cycle of interest and change all visible Code List Views (Thread Windows) to reflect the new cycle of interest
262
End

FIG. 13

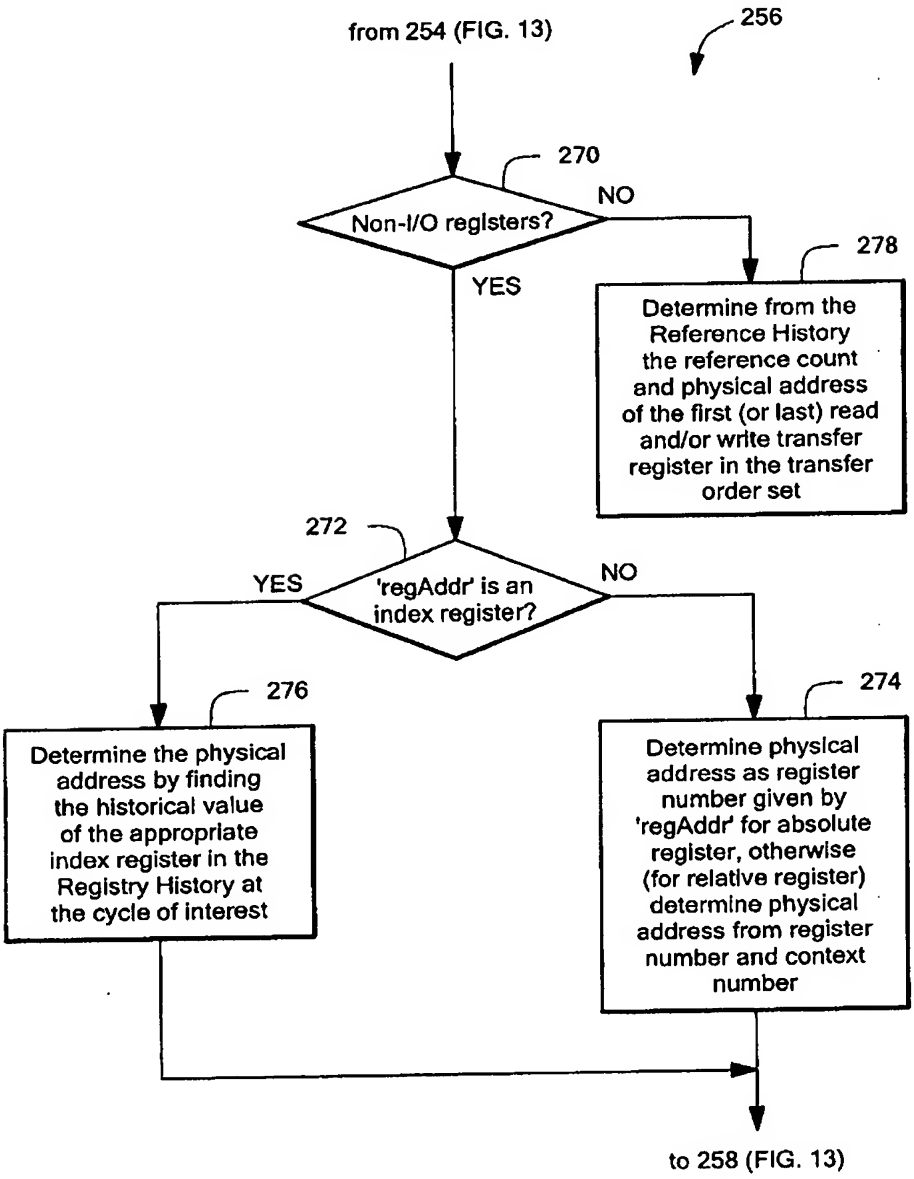
from 254 (FIG. 13)
256
270
Non-I/O registers?
NO
YES
278
Determine from the Reference History the reference count and physical address of the first (or last) read and/or write transfer register in the transfer order set
272
'regAddr' is an index register?
YES
NO
276
Determine the physical address by finding the historical value of the appropriate index register in the Registry History at the cycle of interest
274
Determine physical address as register number given by 'regAddr' for absolute register, otherwise (for relative register) determine physical address from register number and context number
to 258 (FIG. 13)

FIG. 14

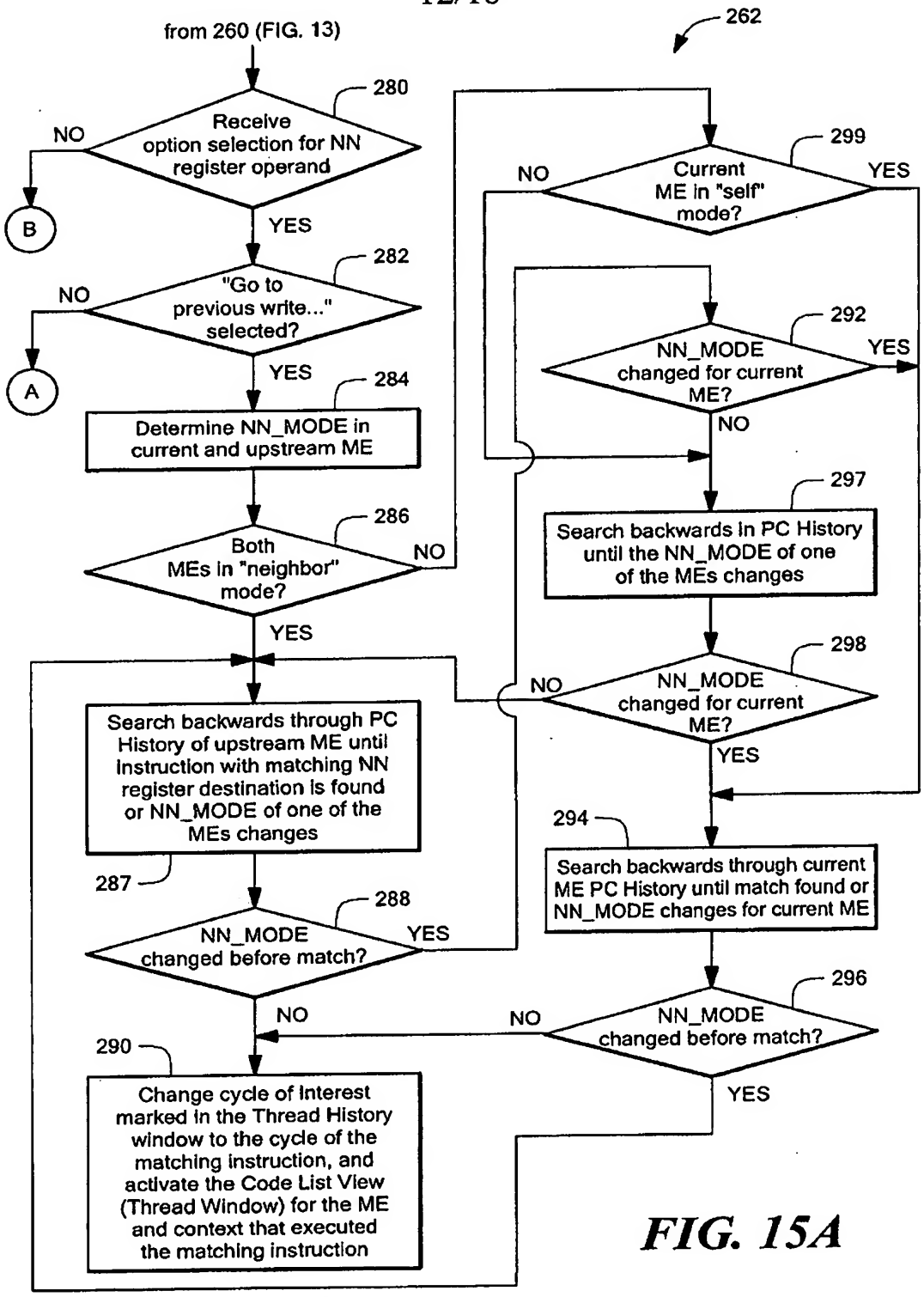
from 260 (FIG. 13)
262
280
Receive option selection for NN register operand
NO
YES
B
282
"Go to previous write..." selected?
NO
YES
A
284
Determine NN_MODE in current and upstream ME
286
Both MEs in "neighbor" mode?
NO
YES
Search backwards through PC History of upstream ME until instruction with matching NN register destination is found or NN_MODE of one of the MEs changes
287
288
NN_MODE changed before match?
YES
NO
290
Change cycle of interest marked in the Thread History window to the cycle of the matching instruction, and activate the Code List View (Thread Window) for the ME and context that executed the matching instruction
299
Current ME in "self" mode?
NO
YES
292
NN_MODE changed for current ME?
YES
NO
297
Search backwards in PC History until the NN_MODE of one of the MEs changes
298
NN_MODE changed for current ME?
NO
YES
294
Search backwards through current ME PC History until match found or NN_MODE changes for current ME
296
NN_MODE changed before match?
NO
YES

FIG. 15A

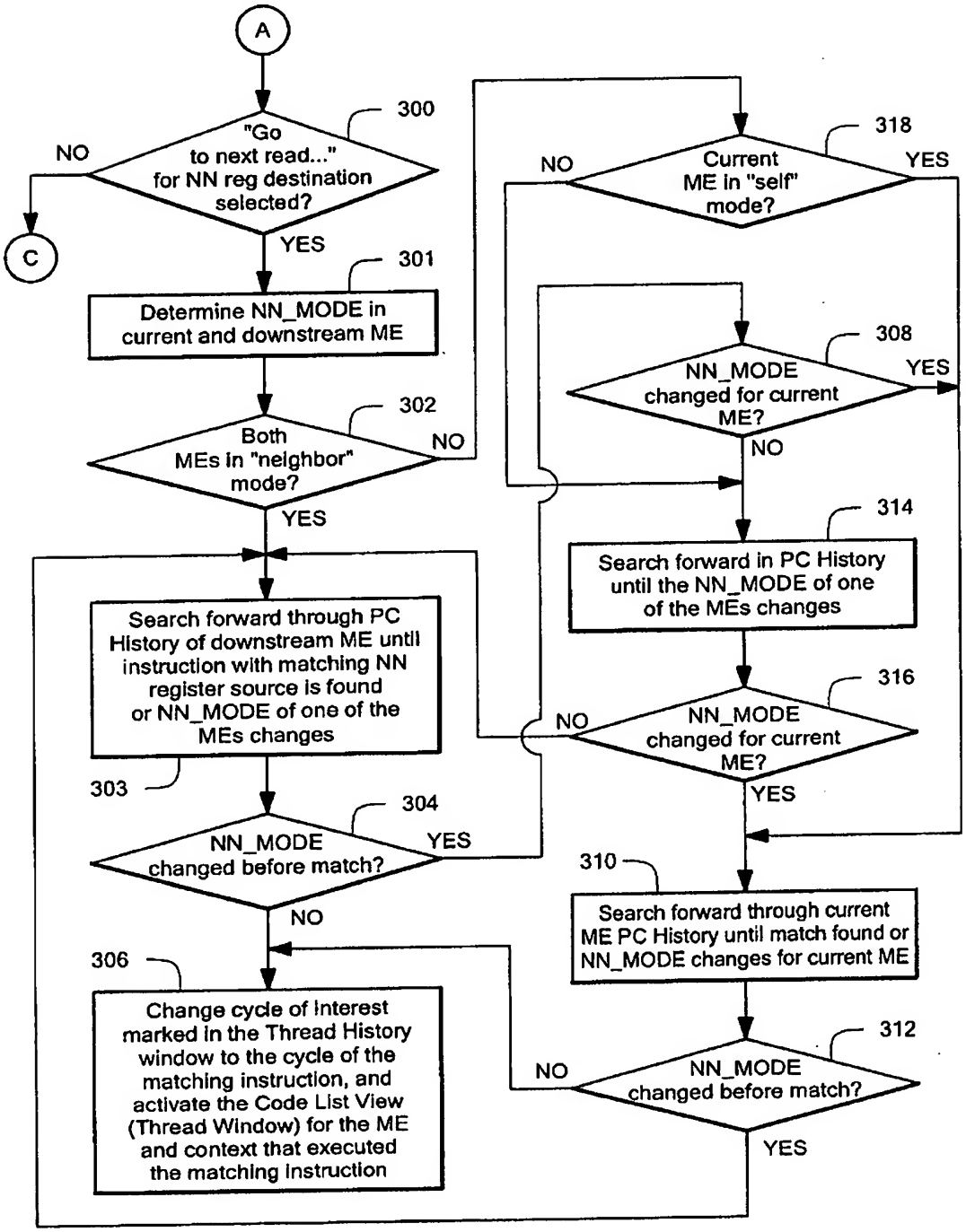
A
"Go to next read..." for NN reg destination selected?
300
NO
YES
C
Determine NN_MODE in current and downstream ME
301
Both MEs in "neighbor" mode?
302
NO
YES
Search forward through PC History of downstream ME until instruction with matching NN register source is found or NN_MODE of one of the MEs changes
303
NN_MODE changed before match?
304
YES
NO
Change cycle of interest marked in the Thread History window to the cycle of the matching instruction, and activate the Code List View (Thread Window) for the ME and context that executed the matching instruction
306
Current ME in "self" mode?
318
NO
YES
NN_MODE changed for current ME?
308
YES
NO
Search forward in PC History until the NN_MODE of one of the MEs changes
314
NN_MODE changed for current ME?
316
NO
YES
Search forward through current ME PC History until match found or NN_MODE changes for current ME
310
NN_MODE changed before match?
312
NO
YES

FIG. 15B

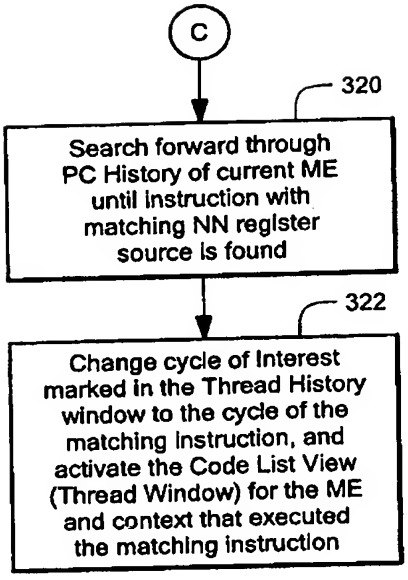

FIG. 15C

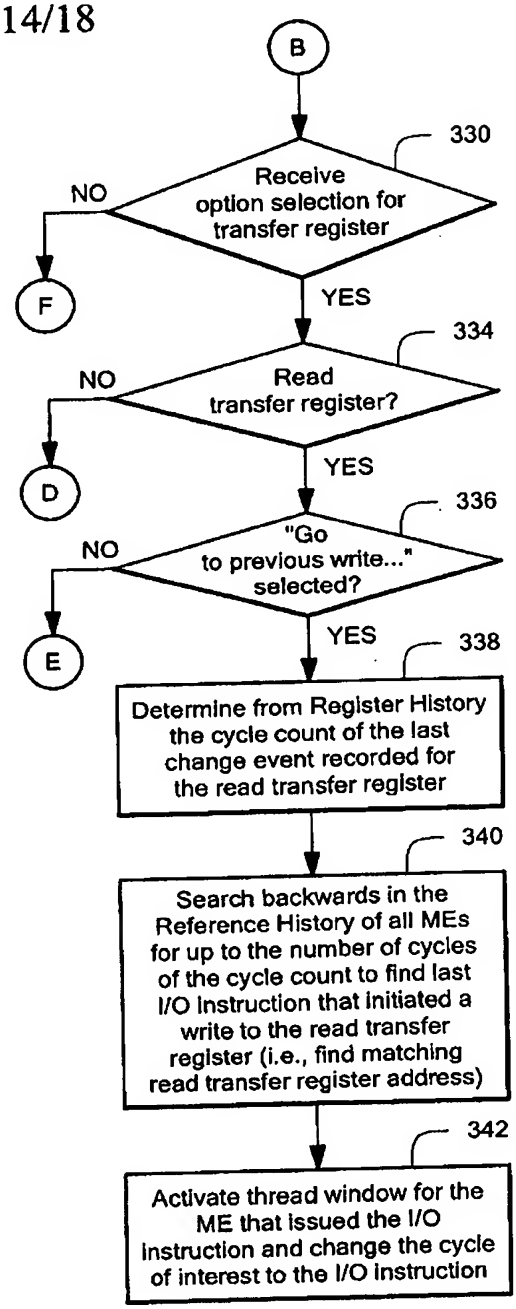

FIG. 15D

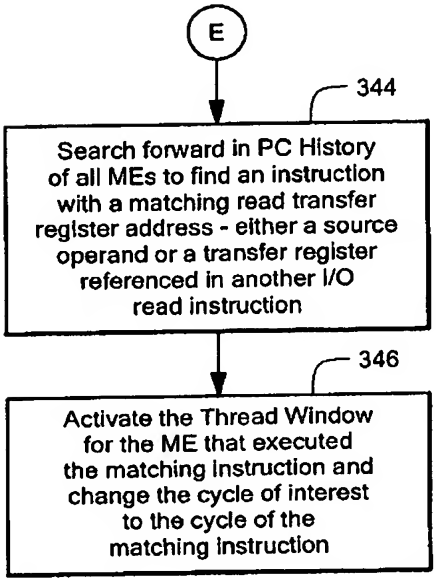
E
344
Search forward in PC History of all MEs to find an instruction with a matching read transfer register address - either a source operand or a transfer register referenced in another I/O read instruction
346
Activate the Thread Window for the ME that executed the matching instruction and change the cycle of interest to the cycle of the matching instruction

FIG. 15E

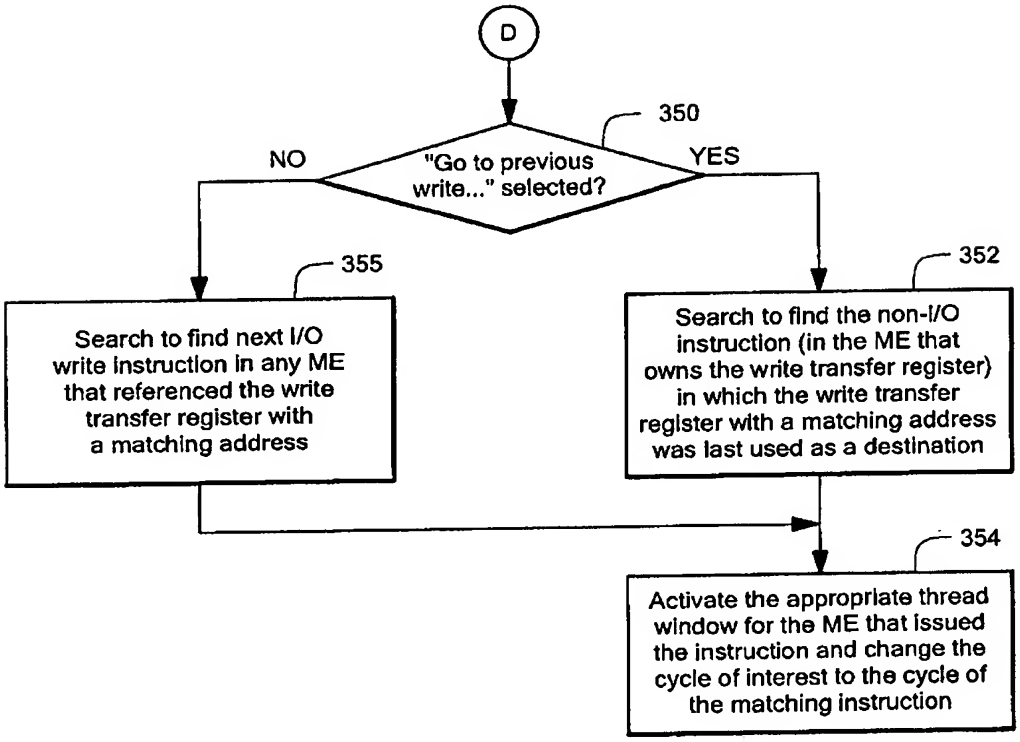
D
350
"Go to previous write..." selected?
NO
YES
355
Search to find next I/O write instruction in any ME that referenced the write transfer register with a matching address
352
Search to find the non-I/O instruction (in the ME that owns the write transfer register) in which the write transfer register with a matching address was last used as a destination
354
Activate the appropriate thread window for the ME that issued the instruction and change the cycle of interest to the cycle of the matching instruction

FIG. 15F

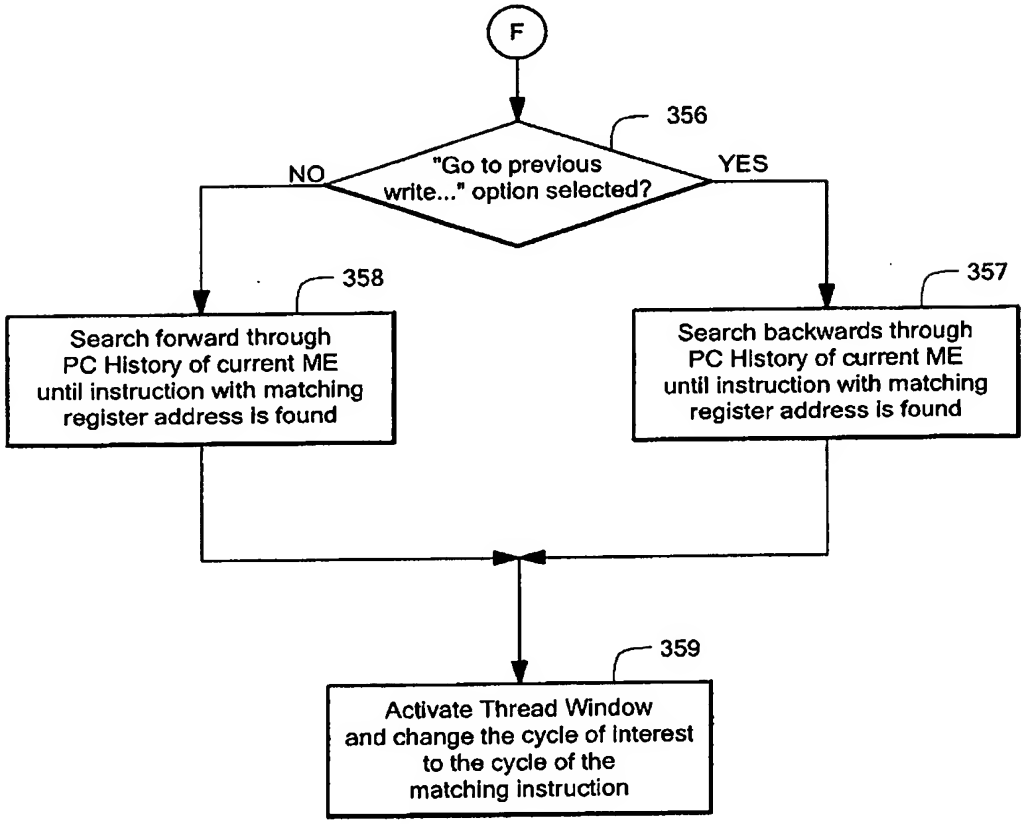
F
356
"Go to previous write..." option selected?
NO
YES
358
Search forward through PC History of current ME until instruction with matching register address is found
357
Search backwards through PC History of current ME until instruction with matching register address is found
359
Activate Thread Window and change the cycle of interest to the cycle of the matching instruction

FIG. 15G

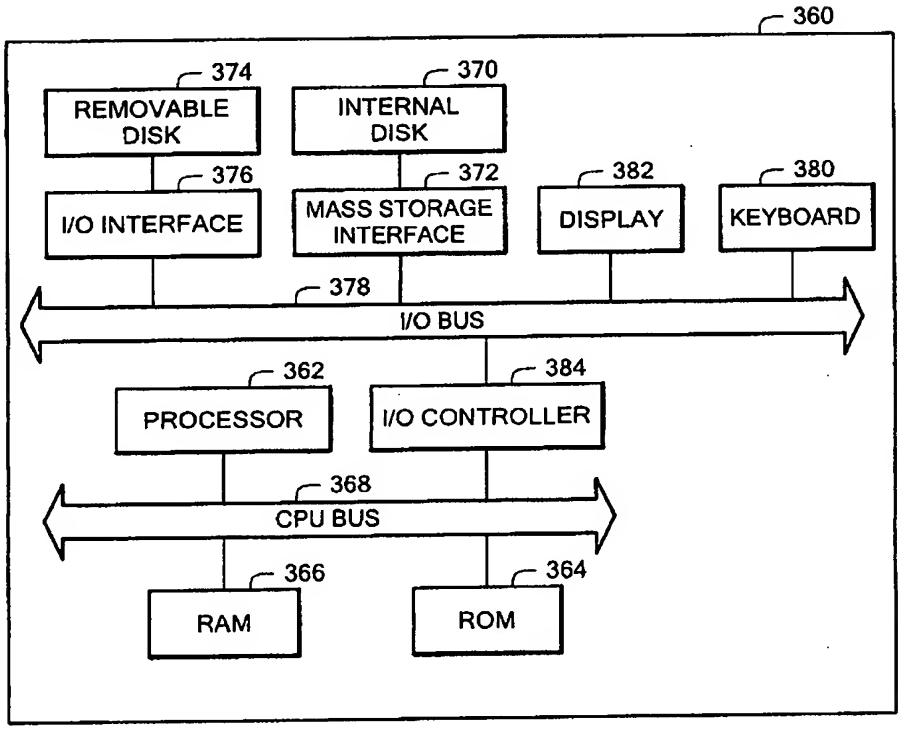
360
374
REMOVABLE DISK
370
INTERNAL DISK
376
I/O INTERFACE
372
MASS STORAGE INTERFACE
382
DISPLAY
380
KEYBOARD
378
I/O BUS
362
PROCESSOR
384
I/O CONTROLLER
368
CPU BUS
366
RAM
364
ROM

FIG. 16